2. 影像處理基礎

2.1 成像原理

- 相機種類

數位相機

USB相機

類比監視相機

可換鏡頭的檢測相機

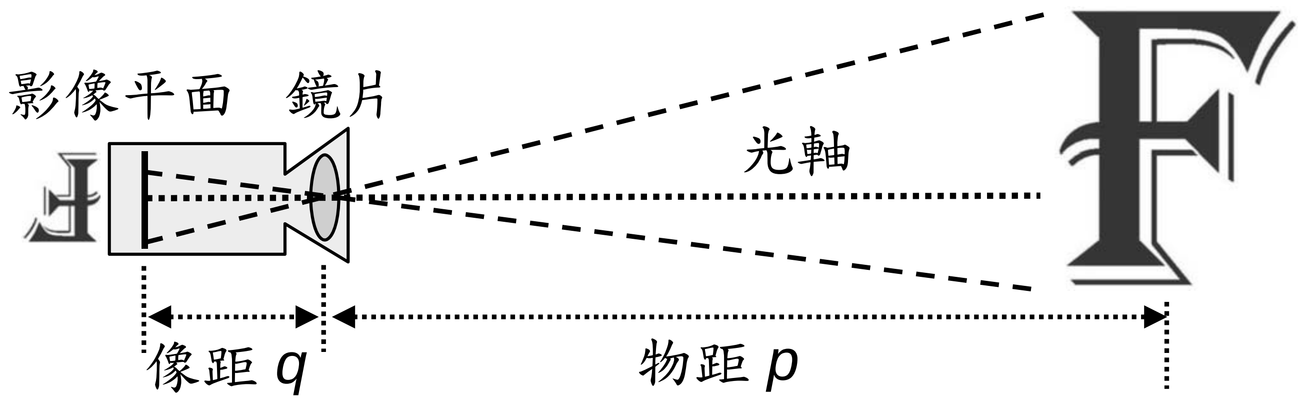
單機板相機

PTZ 相機

全方位相機 (omni-directional)

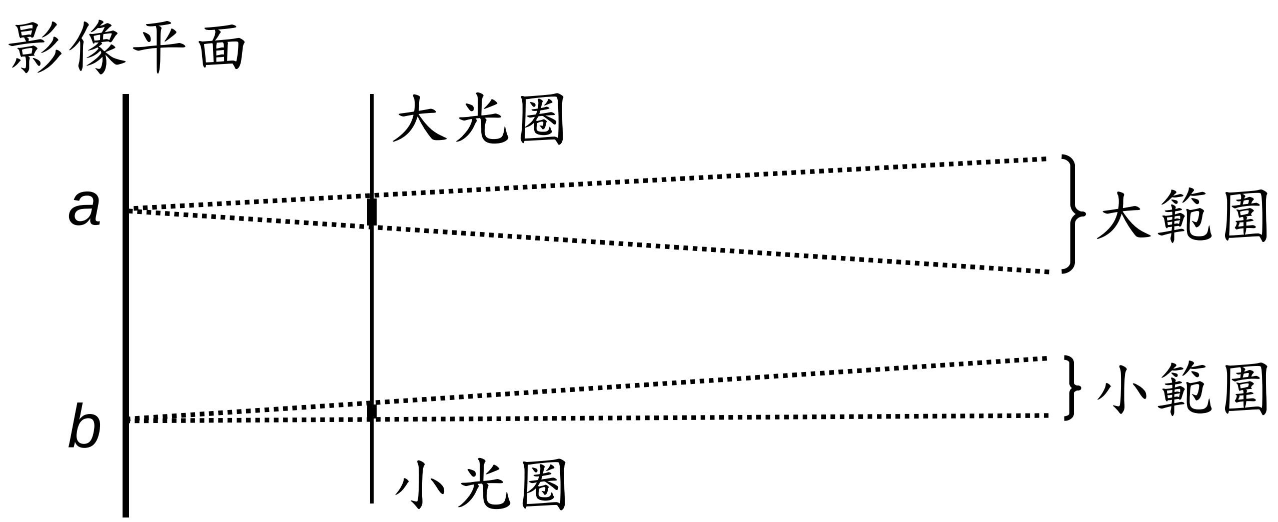
- 針孔成像系統 (pin-hole system) 在光軸 (optical axis) 附近的物體，其清晰成像會滿足 $\frac{1}{f}=\frac{1}{p}+\frac{1}{q}$

  其中 f 是焦距 (focal length)、p 是物距、q 是像距。要特別注意，只有在光軸附近的成像才符合這個公式的關係。

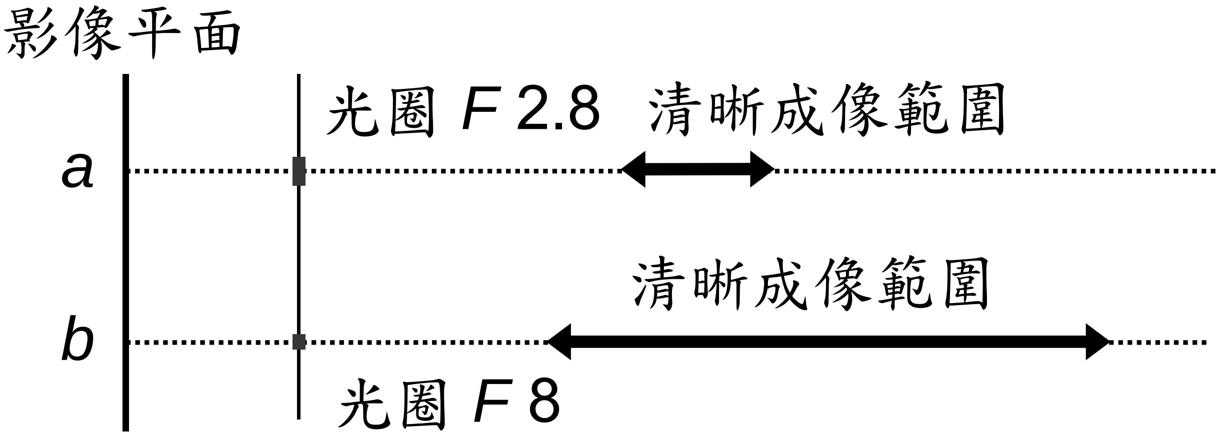


2.1.1 光圈

- 針孔成像系統的針孔大小在攝影上的術語為光圈口徑，簡稱為**光圈** (aperture)。針孔愈大 (光圈愈大，光圈 F 值愈小)，則影像愈模糊 (下圖中 a 點比 b 點模糊)。



- 光圈大小會影響**景深** (depth of field)。景深是指鏡頭前能夠清晰成像的遠近範圍。光圈愈大，景深就愈短。



- 景深範例

景深長的畫面。

景深短的畫面。

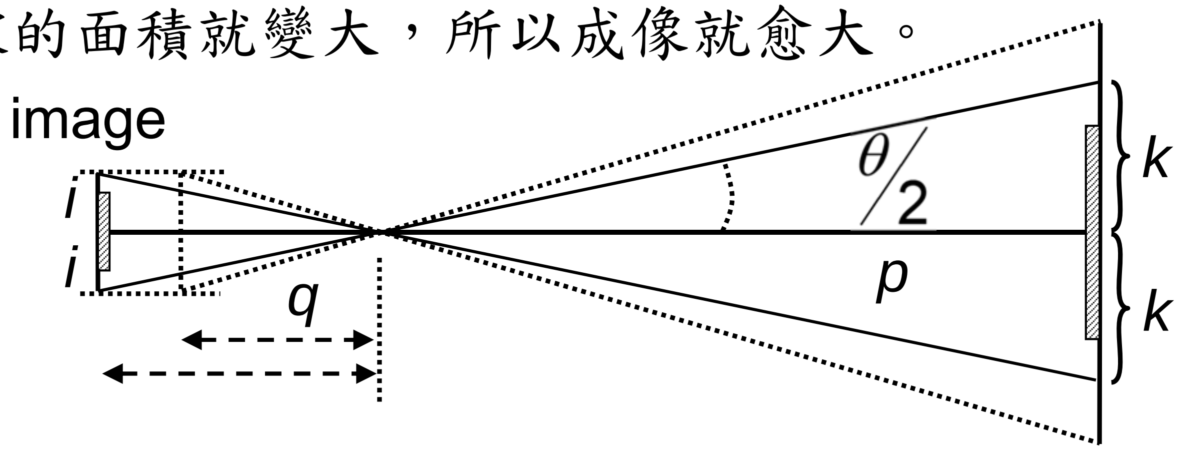
景深短的畫面。

2.1.2 焦距與畫面

- 焦距、照相視角、及畫面放大縮小的關係

鏡頭焦距與視角的關係為 $\tan\frac{\theta}{2} = \frac{i}{q} = \frac{k}{p}$

視角是依底片尺寸及焦距來定義的。(*i*) 焦距固定，則底片愈大，視角就愈大。(*ii*)底片尺寸固定，則焦距愈大，視角就愈小，拍攝物體在畫面中佔據的面積就變大，所以成像就愈大。



- 從上述公式中，我們可以看出，底片尺寸與視角，及焦距與視角的關係，都不是線性關係。

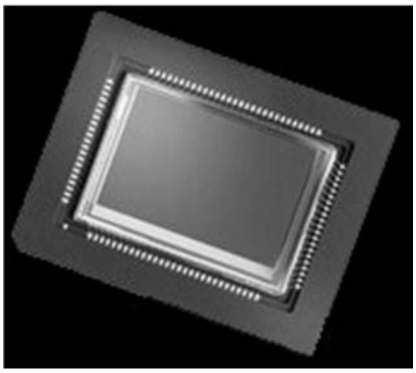
- 上述公式的焦距與視角的關係是要在影像沒有扭曲變形才適用；如果成像有凸變形；例如，使用短焦距鏡頭或魚眼鏡頭，則視角會比正常的大。

- 一般描述視角是對角線方向的視角；但是習慣上，我們常以水平視角來表示我們所看到的範圍。

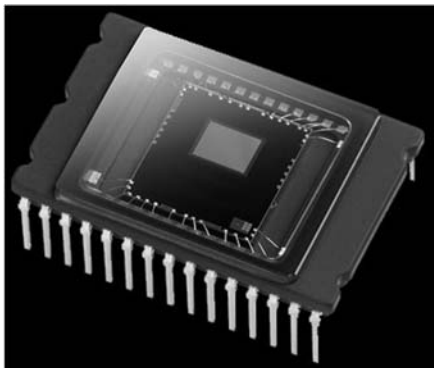
2.2 感光元件

- 人眼之所以能看得到東西，是因為眼睛視網膜上有感光細胞；相機能夠看得到東西，當然也要有感光元件。目前常聽到的電子感光元件有：
  - *i.* 電荷藕合元件 (Charge-Coupled Device, *CCD*)
  - *ii.* 互補性金屬氧化半導體 (Complementary Metal-Oxide Semiconductor, *CMOS*)。

CCD chip



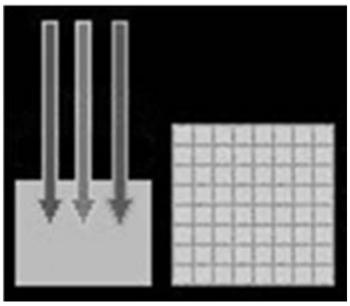
CMOS chip



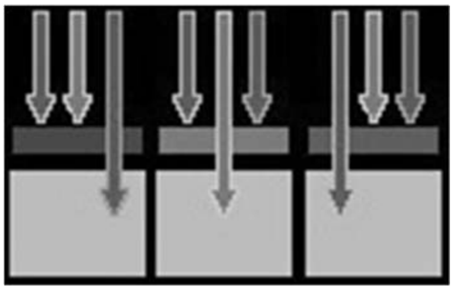
CCD 與 CMOS 的比較

	CCD	CMOS
優點	技術成熟 低雜訊 高感度 線路設計及製程單純	低耗電 低價格 高動態感光範圍 像素可隨機讀取 可與周邊電路一體整合 (*SoC*)
缺點	高耗電量 像素無法隨機讀取 電荷傳遞需接近完美	早期技術不成熟 (現在已成熟) 雜訊較高 晶片線路較複雜

- 人眼可見光波長介於 380 ~ 780 *nm* (奈米)，國際照明協會 (*CIE*) 所訂定的色彩標準：blue: 435.8*nm*, green: 546.1*nm*, red: 700*nm*.

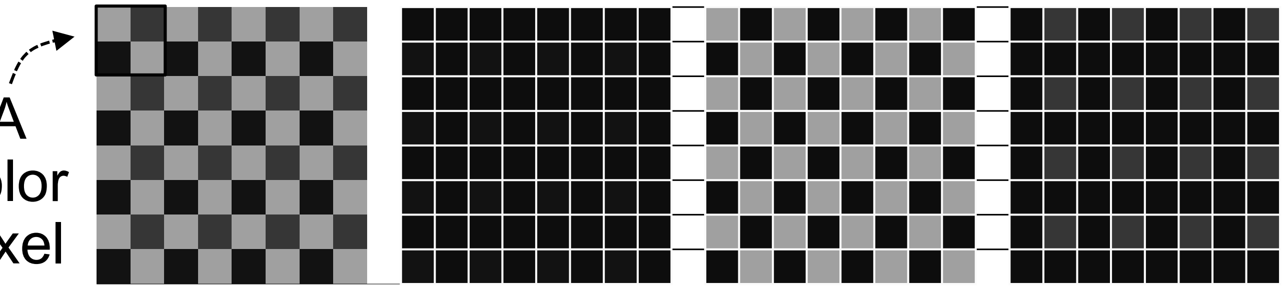


Monochrome sensor



Color sensor

A color pixel



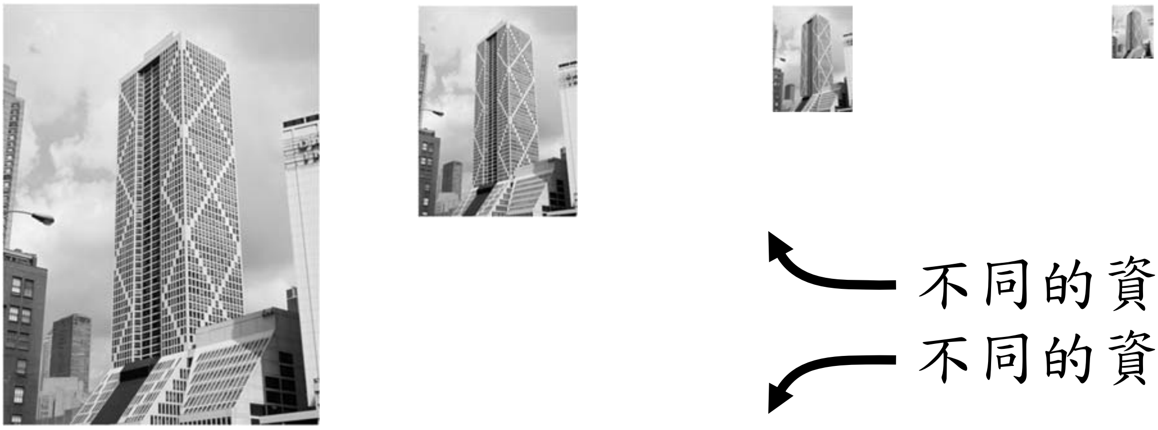
Traditional color filter (Bayer filter)

2.3 抽樣

- 數位化的影像才能做影像處理，而數位化則包含**抽樣** (sampling) 及**量化** (quantization) 兩個動作
- 抽樣是在二度空間的感光元件平面上，水平及垂直兩方向等間距抽取亮度樣本。也可以想像成將一個二維連續亮度函數，在水平及垂直兩個方向個別等間距分割成為許多小方塊，然後計算輸出每一個小方塊的平均亮度。上述的亮度樣本或平均亮度都是實數，要將這些實數轉換成離散的整數，這就是 (灰階) 量化。

- 不同空間解析度影像的比較 (Chicago, America)

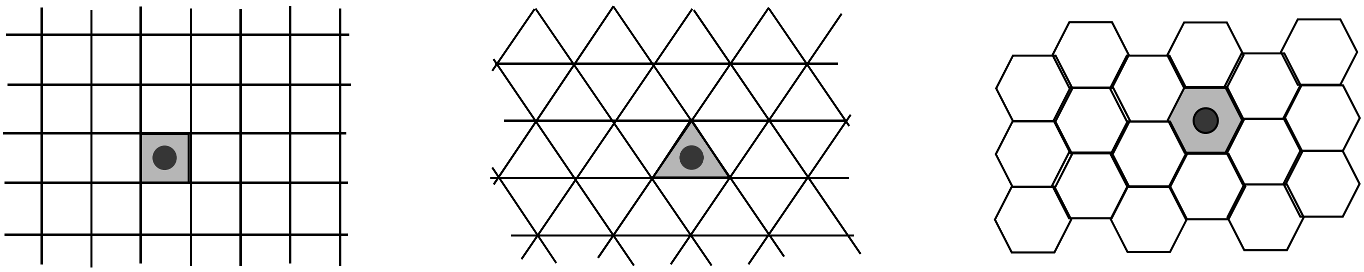
383×512 192×256 96×128 48×64



不同的資訊量

不同的資訊量

- 除了矩形排列的像素外，過去也有三角形 (triangular) 及六角形 (hexagonal) 的排列方式。

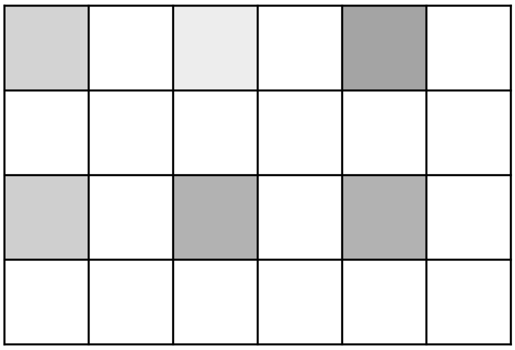


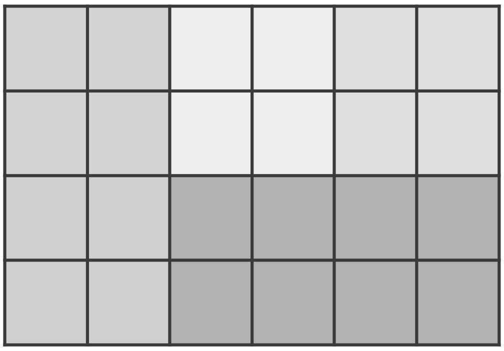
- 抽樣技術中，有一個很有名的理論，稱為“溫德克雪濃抽樣理論” (Whittaker-Shannon sampling theorem)。這個理論說明一個連續性函數，要有怎樣的特性，及在怎樣的抽樣條件下抽樣，可以用抽出的有限樣本重建原始的連續函數。

影像的整數倍縮小

整數倍縮小影像很容易做；例如，要把影像縮小成原來的1/4，則只要把

i. 偶數行及列去掉 或 *ii*. 相鄰4點平均成一點就可以。





影像的整數倍放大

放大影像要補點；例如，2 x 3 放大成 4 x 6；水平及垂直方向各放大一倍，然後在多出來的位置上填入灰階或色彩值。例如，兩個填色方法

i. 用原始影像點複製到相鄰點。

ii. 先用相鄰對角線 4 點的平均值內插 (interpret) 出 *a*、再用原色點及 4 相鄰 *a* 平均內插出 *b*。

1	1	2	2	3	3
1	1	2	2	3	3
4	4	5	5	6	6
4	4	5	5	6	6

	b		*b*		*b*
b	*a*	*b*	*a*	*b*	*a*
	b		*b*		*b*
b	*a*	*b*	*a*	*b*	*a*

- **非整數倍的縮小及放大**都很麻煩。假設輸入影像的大小為 $m_i \times n_i$，輸出影像大小為 $m_o \times n_o$，則輸出影像的每一像素 (x_o, y_o) 的灰階相當於輸入影像之 $(x_o \dfrac{m_i}{m_o}, y_o \dfrac{n_i}{n_o})$ 座標的灰階。但輸入影像中的這個座標不是整數，因此我們要根據這個座標在輸入影像中尋找這個座標附近或包含這個座標的幾個像素的灰階來內插輸出影像之 (x_o, y_o) 像素的灰階。

像素個數 ❹ → ❼

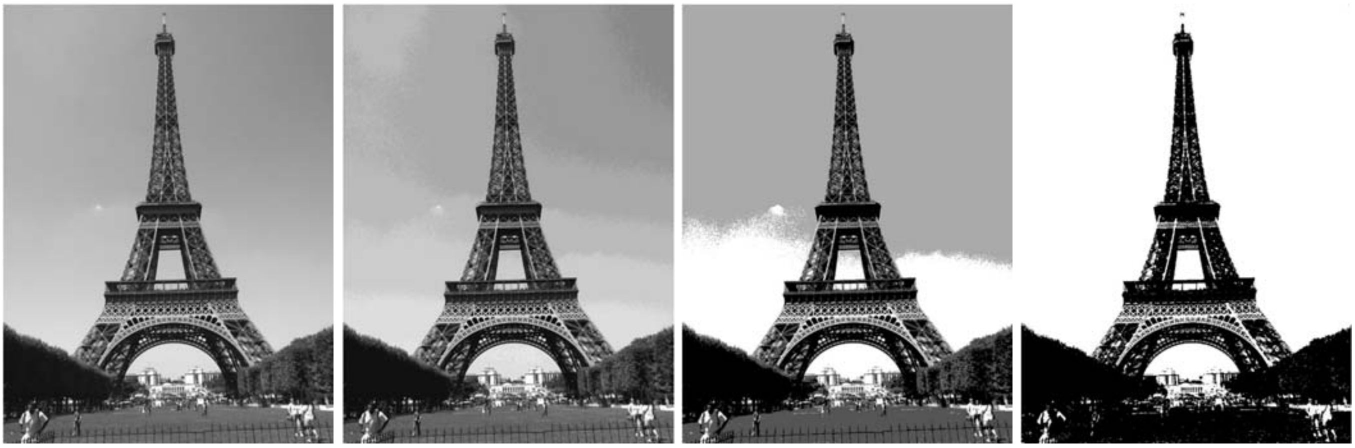
像素位置 $0, 1, 2, 3 \rightarrow 0, \frac{4}{7}, \frac{8}{7}, \frac{12}{7}, \frac{16}{7}, \frac{20}{7}, \frac{24}{7}$

2.4 量化

- **量化 (quantization)** 是當取像設備把光能變成電能後，用數值表示電能 (光能) 的方法。也就是取像設備先做抽樣，接著再用量化給予每一像素一個數值表示該像素的亮度。亮度原來都是用實數表示的，量化後變成了離散整數，所以量化一定會產生量化誤差 (quantization error)。量化的結果稱為**灰階解析度** (gray-level resolution)。同樣的，解析度愈高，影像品質愈好，資料量也愈多

- **解析度** (resolution) 是指描述影像的詳細程度 (the degree of discernible detail)，解析度包含空間解析度及亮度解析度兩個定義。

- 不同灰階解析度影像比較 (Tour Eiffel, Paris, France) .

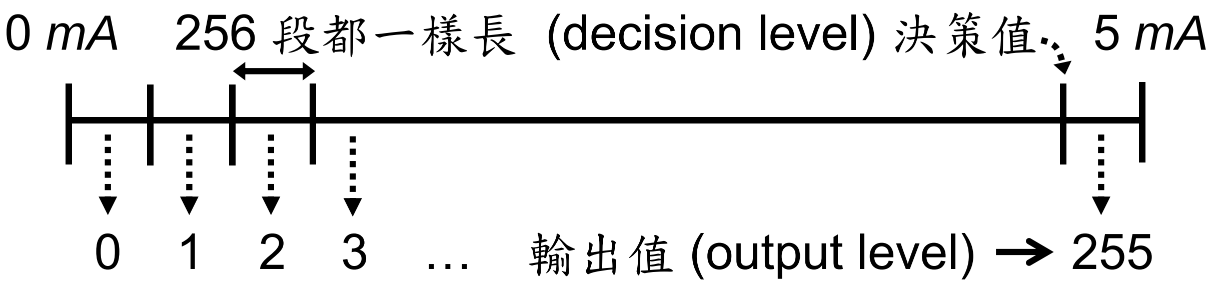


16 levels.　8 levels.　4 levels.　2 levels.

- 本節將介紹三種量化技術：一種均勻量化、二種調適性量化。

2.4.1 均勻量化

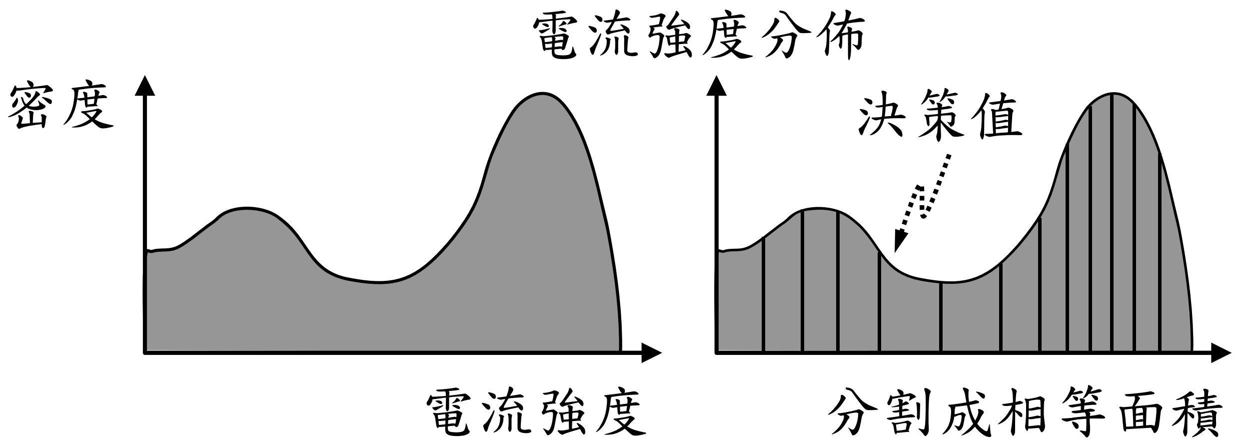
- **均勻量化**又稱為**等間距量化** (equally-spaced quantization)。
- 用以分割量化區間的數值稱為決策值 (decision level)，而每一段落給予一個數值表示一個特定的灰階稱為輸出值 (output level)。



The output level *i* is located at the midpoint of an interval.

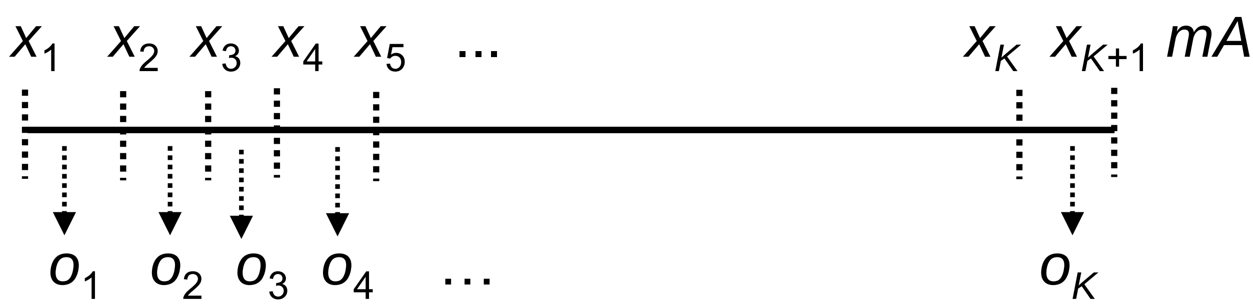
2.4.2 漸縮量化

- 漸縮量化 (tapered quantization) 是一種調適性量化 (adaptive quantization)。它的意義是根據所有感光元件的輸出電流強度分佈 (distribution) 來決定量化決策值的位置。分佈比較密集的地方分割細一點 (也就是決策值較密集)，分佈比較稀少的地方分割粗一點 (決策值較稀疏)



2.4.3 最佳量化

- 最佳量化 (optimal quantization)，顧名思義就是最好的量化；但所謂最好是針對數學公式而言，不是說眼睛看起來最好的意思。數學公式的意義不一定與眼睛的感受一致，所以對數學公式的結果最好，對眼睛卻不一定是最好的。眼睛看起來好不好常帶有心理的因素，因此人眼視覺的感受並不容易定義。



- 在每一量化段落中，本來都有各種不同的實數輸出電流強度，量化是要把這些不同的電流強度都量化成相同的整數輸出值。以 $[x_i, x_{i+1}]$ 這個段落為例，輸出值為o_i，所以這個段落的量化平方誤差 (mean-square quantization error) 為

$$\int_{x_i}^{x_{i+1}} (x-o_i)^2 p(x)\, dx$$

其中$p(x)$ 是量化前具有 x 亮度之像素的機率值。整個量化共有K個段落，且各段落的誤差都是獨立的，所以總量化平方誤差 δ_q^2 即為 K 段落誤差的總和

$$\delta_q^2 = \sum_{i=1}^{K} \int_{x_i}^{x_{i+1}} (x-o_i)^2\, p(x)\, dx$$

- 如何在 δ_q^2 最小的情形下，求出2K-1未知的輸出值及決策值？一般求極值的方法就是以未知變數對誤差函數做一次微分為零的方程式中求未知變數值。

$$\frac{\partial\, \delta_q^2}{\partial\, x_i} = (x_i - o_{i-1})^2\, p(x_i) - (x_i - o_i)^2\, p(x_i) = 0, \quad i = 2, 3, ..., K.$$

$$\frac{\partial\, \delta_q^2}{\partial\, o_i} = -2 \int_{x_i}^{x_{i+1}} (x - o_i)\, p(x)\, dx = 0, \quad i = 1, 2, ..., K.$$

假設所有機率值 $p(x_i)$ 都不為 0

$$x_i = \frac{o_{i-1} + o_i}{2}, i = 2, 3, .., K. \qquad (1)$$

$$o_i = \frac{\int_{x_i}^{x_{i+1}} x\, p(x)\, dx}{\int_{x_i}^{x_{i+1}} p(x)\, dx}, i = 1, 2, ..., K. \qquad (2)$$

- 上述公式的意義是
  - *i*. x_i 決策值落在二個相鄰輸出值 $[o_i, o_{i+1}]$ 中間
  - *ii*. oi 輸出值永遠落在二個相鄰決策值 $[x_i, x_{i+1}]$ 的質量中心 (centroid) 位置。

(*a*) 均勻量化 (uniform quantization)

如果 $p(z)$ 是均勻分佈，$o_i = \dfrac{\frac{1}{2}\left(x_{i+1}^2 - x_i^2\right)}{x_{i+1} - x_i} = \dfrac{x_{i+1} + x_i}{2}$

則 $x_i = \dfrac{o_{i-1} + o_i}{2}$, $i = 2, 3, \ldots, K$.

$o_i = \dfrac{x_i + x_{i+1}}{2}$, $i = 1, 2, \ldots, K$.

That is, x_1, o_1,... , x_K, o_K, x_{K+1} are all evenly spaced.

(*b*) 非均勻量化 (non-uniform quantization)

如果 $p(z)$ 是非均勻分佈，則我們只能應用嘗試錯誤 (try and error) 的策略來解未知數 x_i's 及 o_i's。

處理程序

*step*1. Pick arbitrary o_1 value and by Eqs.(2) and (1) from known z_1, o_1 and $p(z)$ to solve z_2 and o_2.

*step*2. Then x_3, o_3; x_4, o_4; ... all can be solved while finally x_K and o_K are solved.

*step*3. Check whether $o_K = \int_{x_K}^{x_{K+1}} xp(x)dx \Big/ \int_{x_K}^{x_{K+1}} p(x)dx$ is satisfied.

*step*4. If not, try a new o_1 among its possible range until *step*3 is satisfied to some degree.

End.

- 討論
- 如何寫程式來完成上述的演算法，要包含有系統的自動調整 o_1，要根據什麼誤差來調整o_1？這要用腦筋想想，o_1調整得好，重複演算的次數會較少，速度就較快。
- 最佳量化與漸縮量化的觀念是不一樣的，但他們的結果卻有一個共同點，就是在機率密度較高的地方，切割出來的段落會較窄 (量化成多個灰階)，在機率密度較低的地方，切割出來的段落就比較寬 (量化成較少個灰階)。

2.4.4 抽樣與量化的抉擇

- "好影像" 很難定義的，因為影像好壞與取相設備、影像內容、環境、及個人主觀判定等因素相關
- 通常是空間解析度及色彩解析度愈高，影像品質就愈好。
- 但是解析度愈高，資料量也愈多，處理時間、儲存空間、及傳輸時間也跟著增加。如何在固定影像資料量下，取捨空間解析度及色彩解析度？
  i. 對於簡單影像而言，色彩解析度要高，空間解析度可以低一些。
  ii. 對於複雜影像而言，空間解析度要高一點，色彩解析度可以低一些。

- 三種不同複雜度的影像

簡單影像。 普通影像。 複雜影像。

2.5 像素間的關係

- 像素間的幾何及拓墣關係 (geometric and topology relationship) 包括：相鄰、相連、區塊、距離等關係。
- 在描述像素的相連與區塊關係時，要將類似灰階的像素連結在一起是很難的工作，因為 “類似” 很難定義。所以我們都會先將像素二值化 (bi-level thresholding)；也就是先將像素分成兩類，一類是要處理的物體內部像素，習慣用 “1” 表示，稱為前景點 (foreground pixel)，一類是不被處理的背景像素，習慣用 “0” 表示，稱為背景點 (background pixel)。

2.5.1 相鄰像素 (neighbors of a pixel)

- 像素 p 的相鄰 (neighbor) 點定義

  - p 的四鄰點  ，表示為 $N_4(p)$.

  - p 的對角線鄰點  ，表示為 $N_D(p)$.

  - p 的八鄰點  ，表示為 $N_8(p)$.

2.5.2 像素的相連性 (connectivity)

- **目的**

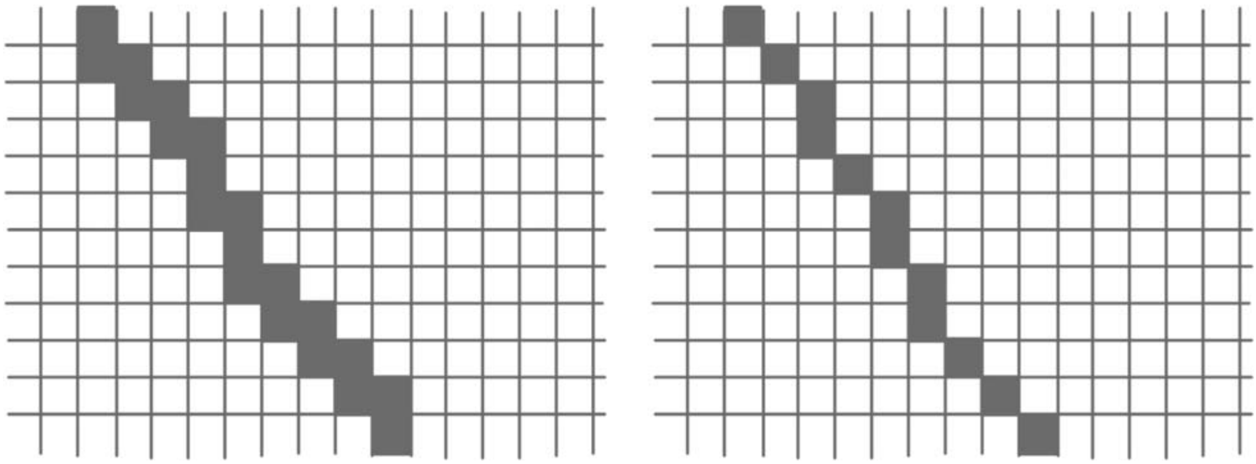
  以一條連續封閉曲線把屬於一個物體的像素包圍起來；而所謂的連續曲線必然是曲線上的相鄰像素一定要**相連** (connect)。

- 令 $\boldsymbol{V}$ 是前景點集合，也就是定義邊界 (edge) 的點集合。假設有兩個前景點 p 及 q：

  i. 如果 q 是在 $\boldsymbol{N}_4(p)$ 集合內，則稱 p 及 q 是 **4相連** (4-connected)，

  ii. 如果 q 是在 $\boldsymbol{N}_8(p)$ 集合內，則稱 p 及 q 是 **8相連** (8-connected)。

- 不同定義的相連曲線



4-相連曲線　　8-相連曲線

- **兩區塊鄰接 (adjacent) 的定義**
  影像中的一個區塊相當於一些相連點所形成的一個點集合。若 S_1 區塊中有一個點與 S_2 區塊中的某一點相連 (connect)，則稱 S_1 區塊與 S_2 區塊相鄰接 (adjacent)。

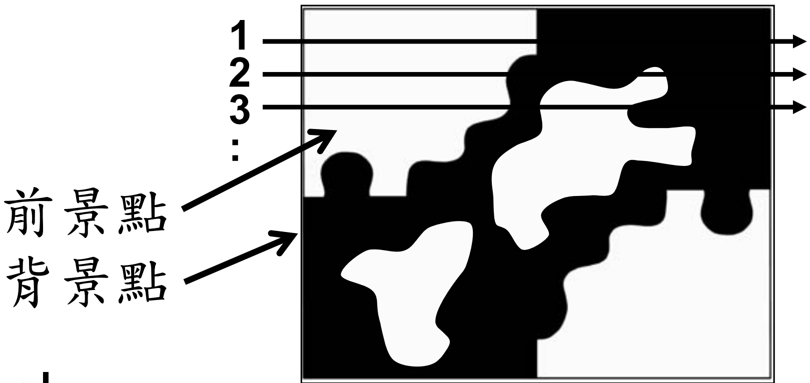
- 相連有 4-相連與 8-相連兩種定義，根據相連點的定義；S_1 與 S_2 區塊也有4-相鄰接 (4-adjacent) 與 8-相鄰接 (8-adjacent) 兩種定義。

2.5.3 連結區塊的產生

- 目的：以像素標記 (labeling) 的方法給予相連且類似灰階的像素相同的標記 (label)。之後可以將具有相同標記的像素連成一個區塊，稱為**連結區塊** (connected component)。
- 產生連結區塊演算法的特色：
  - 兩階段步驟 (two passes)，
  - 僅適用於二值化影像，1是前景點，0是背景點

A. 4-連結區塊

	r	
t	*p*	



pass 1: (scanning)

if p = 0, scan next pixel,

if p = 1, if $r = t = 0$, set p a new label,

if r = 1 and t = 0, set $L(p) = L(r)$,

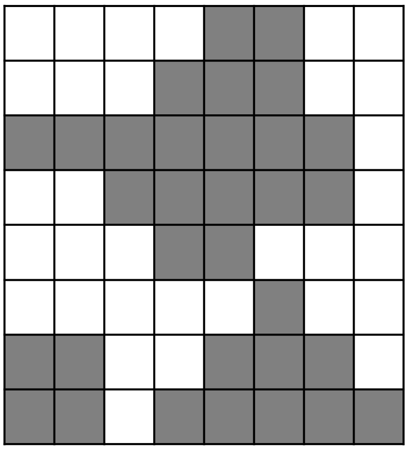
if r = 0 and t = 1, set $L(p) = L(t)$,

if r = 1 and t = 1, and $L(r) = L(t)$, set $L(p) = L(r)$ or $L(t)$

if r = 1 and t = 1, and $L(r) \neq L(t)$, set $L(p) = L(r)$ and mark $L(t) = L(r)$.

pass 2: (merging classes) merge "$L(t) = L(r)$" classes.

✿ 4-連結區塊範例



二值化影像。

				1	1		
			2	1	1		
3	3	3	2	1	1	1	
		3	2	1	1	1	
			2	1			
					4		
5	5			6	4	4	
5	5		7	6	4	4	4

第一階段標記。

				1	1		
			1	1	1		
1	1	1	1	1	1	1	
		1	1	1	1	1	
			1	1			
					4		
5	5			4	4	4	
5	5		4	4	4	4	4

第二階段標記。

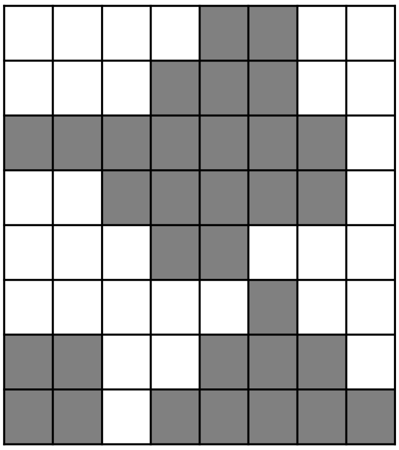
B. **8-**連結區塊

q	r	s
t	p	

8-連結區塊產生的方法與
4-連結區塊完全相同，
只是判定準則多幾個。

pass 1: (scanning)
if p = 0, scan next pixel.
if p = 1, if all $\{q,r,s,t\}$ = 0, set p a new label,
 if only one of $\{q,r,s,t\}$ = 1,
  set $L(p) = L(x)$, where x = 1 and $x \in \{q,r,s,t\}$.
 if more than one of $\{q,r,s,t\}$ = 1,
  set $L(p) = L(x)$, where x = 1, $x \in \{q,r,s,t\}$, and
  mark $L(y) = L(x)$ for all other y = 1 and $y \in \{q,r,s,t\}$.
pass 2: (merging classes) merge all equivalent classes.

8-連結區塊範例



二值化影像。

				1	1		
			1	1	1		
2	2	1	1	1	1	1	
		1	1	1	1	1	
			1	1			
					1		
3	3			1	1	1	
3	3		1	1	1	1	1

第一階段標記。

				1	1		
			1	1	1		
1	1	1	1	1	1	1	
		1	1	1	1	1	
			1	1			
					1		
3	3			1	1	1	
3	3		1	1	1	1	1

第二階段標記。

4-連結區塊 與 **8-**連結區塊 哪一個比較好？

比較 效果 (effect) 與 效率 (efficiency)

2.5.4 像素間的距離 (distance measurement)

- 假設影像中有兩像素*p* 及 *q*，其直角座標系統的座標各為 (*x*, *y*) 及 (*s*, *t*)，則 *p*, *q* 二點的

  i. 歐基理德距離 (Euclidean distance) 定義為

  $$D_e(p, q) = \sqrt{(x-s)^2 + (y-t)^2}$$

  ii. 城市區塊距離 (city-block distance) 定義為

  $$D_4(p, q) = |x-s| + |y-t|$$

  iii. 棋盤距離 (chessboard distance) 定義為

  $$D_8(p, q) = max(|x-s|, |y-t|)$$

$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
2	1	0	1	2
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$

歐基理德距離

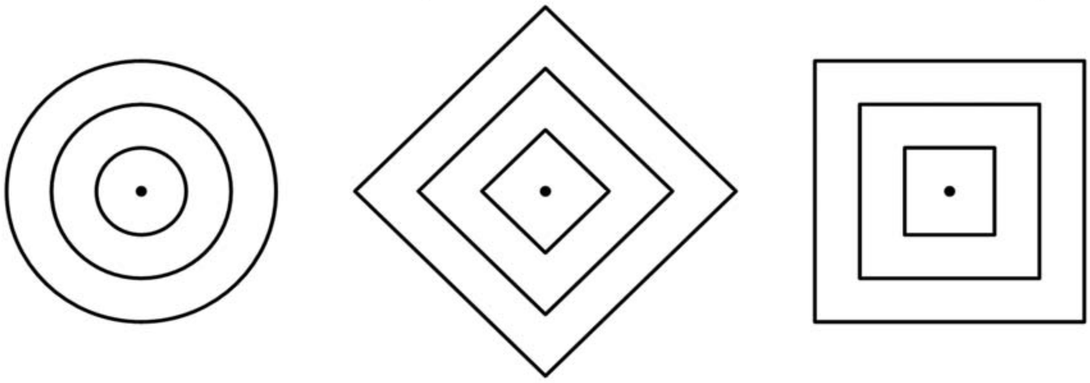
4	3	2	3	4
3	2	1	2	3
2	1	0	1	2
3	2	1	2	3
4	3	2	3	4

城市區塊距離

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

棋盤距離

- 至中心點的等距線 (circum-distance curve)



2.5.5 像素間的運算 (operations of image processing)

- 我們將影像處理的像素運算 (operations between pixels) 根據運算影響範圍，分成三個層次 (levels of operations)：
  - *i.* 點運算 (point operation)
  - *ii.* 區域運算 (local operation)
  - *iii.* 全運算 (global operation)

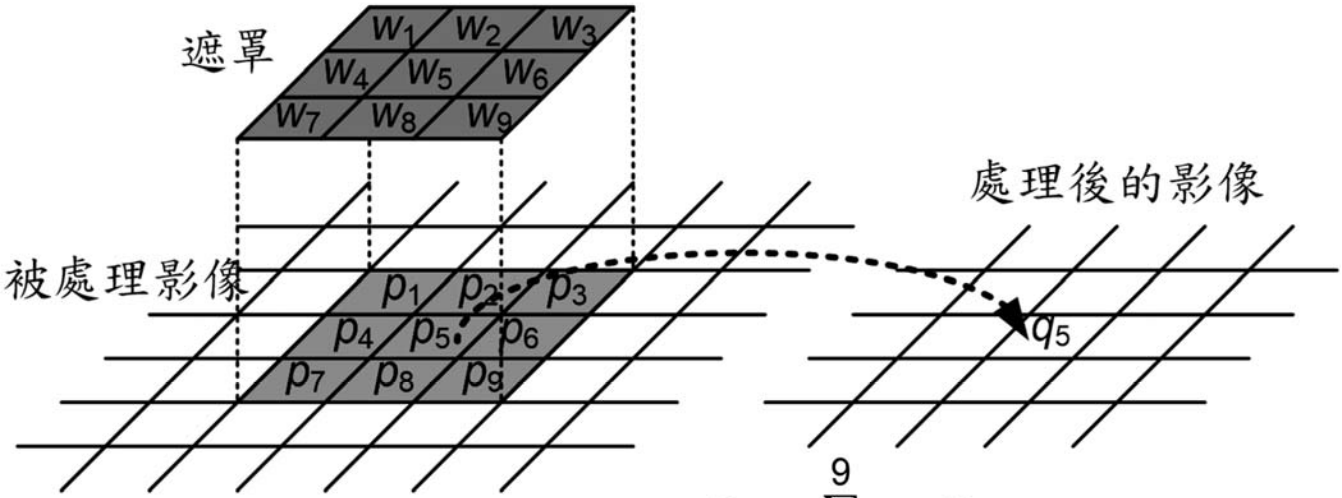
A. 點運算

- 一個像素一個像素的處理方式稱為點運算，其重點在於處理某一像素時，與其他像素無關。
- 我們可以將點運算分成兩類，一類是改變像素的灰階或色彩值不受其他像素的影響；例如，強化對比、灰階二值化、擴大灰階尺度、像素分群 (clustering)、.. 等處理。
- 另一類是改變像素在影像中的座標值；例如，位移、旋轉、縮放、變形等的幾何轉換 (geometric transformation)。雖然每一像素都做相同的轉換，但每一像素的轉換都不影響其他像素的轉換，也不受其他像素轉換的影響。

B. 區域運算

- 區域運算是一個區域一個區域的處理方式，但每一區域運算只有一個像素的灰階或色彩值會改變；也就是某一像素改變灰階或色彩值時，會受到區塊內其他像素灰階或色彩值的影響，這種處理方式就稱為區域運算 (local operation)。
- 所以在區域運算中，某一像素的灰階或色彩值改變會受到週邊像素的影響。在所有運算中，區域運算是最重要的。

- 可以應用的方向有：強化對比、去除雜訊、平滑化、銳利化、特徵擷取、骨架化、.. 等。
- 最常用的區域運算方式是 遮罩運算 (mask operator)



$$q_5 = \sum_{j=1}^{9} w_j p_j$$

- 遮罩大小可以為 3×1, 1×3, 3×3, 5×5, 7×7, …

w_1
w_2
w_3

3×1 遮罩。

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

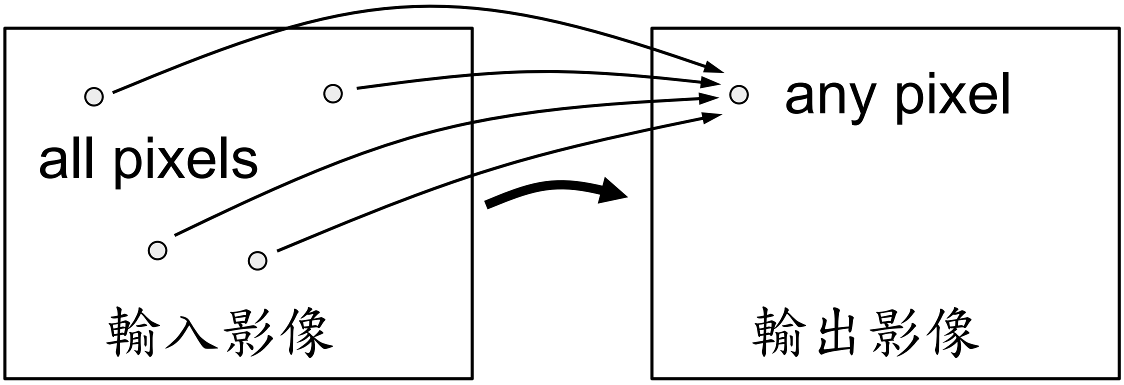
3×3 遮罩。

w_1	w_2	w_3	w_4	w_5
w_6	w_7	w_8	w_9	w_{10}
w_{11}	w_{12}	w_{13}	w_{14}	w_{15}
w_{16}	w_{27}	w_{18}	w_{19}	w_{20}
w_{21}	w_{22}	w_{23}	w_{24}	w_{25}

5×5 遮罩。

C. 全體運算

- 全體運算的意義是說一個像素的某種數值計算會受到整張影像所有像素之灰階或色彩值的影響；例如，傅立葉轉換，每一個像素的數值計算都會用到整張影像中的每一個像素；也就是每個一轉換值都與每一像素相關。其他屬於全運算的轉換還有，離散餘弦轉換、.. 等。



2.6 影像幾何轉換

- 影像幾何轉換 (image geometric transformation) 大都是用來修改 (modify) 或修正 (correct) 影像像素間的空間關係；例如，影像位移 (translation)、縮放 (scaling)、旋轉 (rotation)、斜變 (skew or shear transform)、或扭曲變形 (distortion) 等二維到二維的基本轉換。

- 一個二維影像幾何轉換通常都包括兩個基本動作

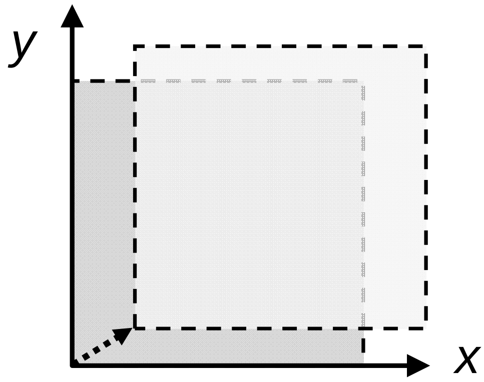
  i. 二維空間轉換 (spatial coordinate transformation)

  ii. 灰階內插 (gray-level interpolation)。

2.6.1 二維基本幾何轉換

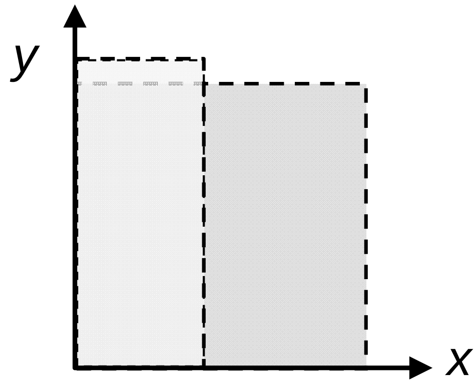
$\begin{bmatrix} x \\ y \end{bmatrix}$ 舊座標，$\begin{bmatrix} x' \\ y' \end{bmatrix}$ 新座標

A. 位移 (translation)



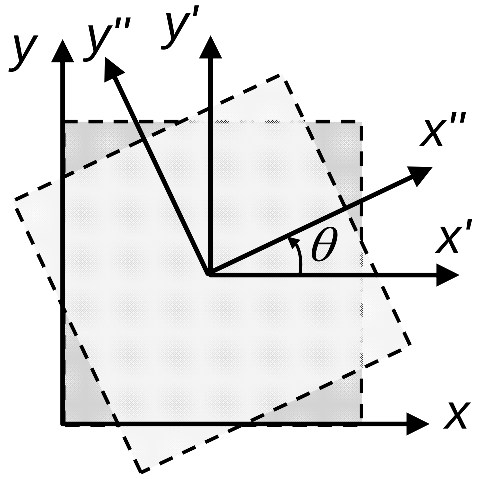
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

B. 縮放 (scaling)



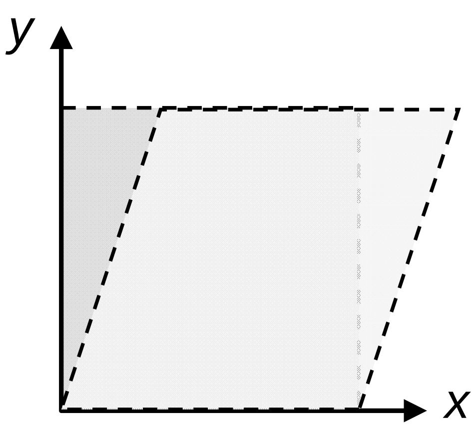
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

C. 旋轉 (rotation)



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

D. 斜變 (skew or shear transform)



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \text{ or}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

E. 歐機里德轉換

(Euclidean trans.
= Rigid trans.)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

There are three parameters:
θ, t_x, and t_y (3 *DOF*).

F. 相似轉換

(similarity trans.)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta \\ s\sin\theta & s\cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

There are three parameters:
s, θ, t_x, and t_y
(4 *DOF*).

G. 仿射轉換

(Affine transform.)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

There are six parameters:
a_{11}, a_{12}, a_{21}, a_{22}, t_x, and t_y (6 *DOF*).

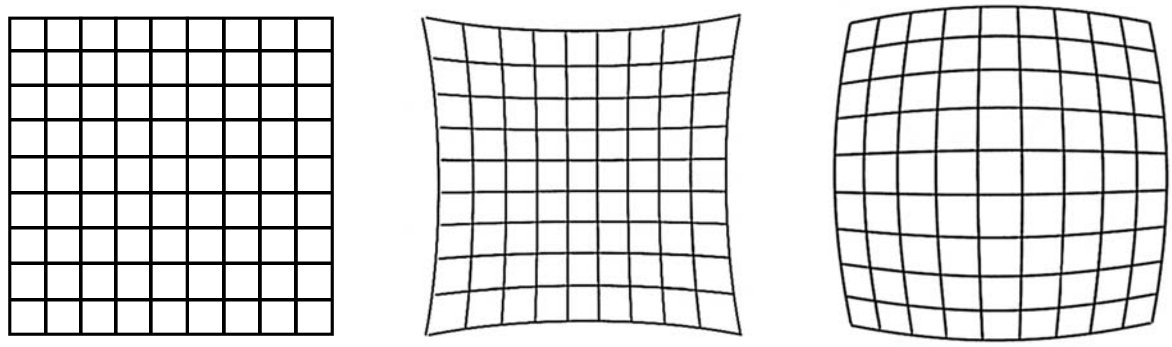
H. 投影轉換

(projective trans.)

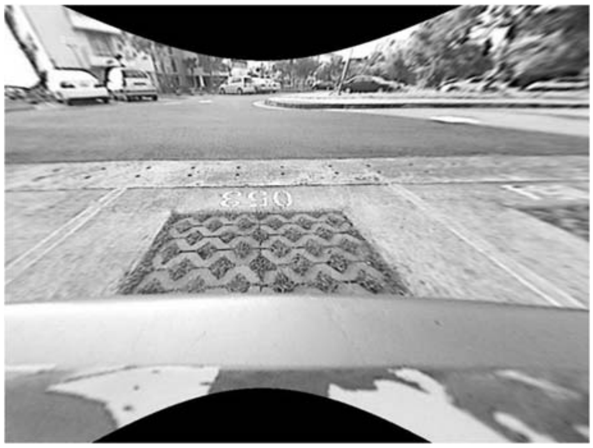
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ x \end{bmatrix}$$

This is a 3D to 2D transformation.
There are eight parameters:
h_{ij}'s (8 *DOF*).

2.6.2 二維扭曲轉換 (distortion calibration)



無扭曲畫面。針墊扭曲畫面。酒桶扭曲畫面。



- 假設正常影像 $f(x, y)$ 經過 $x' = T_x(x, y)$ 及 $y' = T_y(x, y)$ 的變形後，變成扭曲影像 $g(x', y')$。要從扭曲影像重建正常影像要先知道變形的數學公式。如果變形公式未知，則要先估計出變形公式。估計變形公式有兩個步驟

  i. 建立數學模式 (model)，

  ii. 估計模式中的參數 (parameter)。

  數學模式可定義為

$$\begin{cases} T_x(x,y) = c_1 + c_2 x + c_3 y + c_4 xy \\ T_y(x,y) = c_5 + c_6 x + c_7 y + c_8 xy \end{cases} \tag{3}$$

  其中 c_i, $i = 1, 2, ..., 8$ 是待估計的未知參數。

- 上述公式是有限制的曲面關係，校正效果有限；因此我們將兩公式各加入 x, y 變數的 2 次方項，變成

$$\begin{cases} T_x(x,y) = c_1 + c_2 x + c_3 y + c_4 xy + c_5 x^2 + c_6 y^2 \\ T_y(x,y) = c_7 + c_8 x + c_9 y + c_{10} xy + c_{11} x^2 + c_{12} y^2 \end{cases} \quad (4)$$

其中 c_i, i = 1, 2, ..., 12 是未知參數。

- 理論上，估計項數越多，越準確；我們也可以將估計項數增加到 10 項，包含 x, y 變數的 3 次方項，

$$\begin{cases} T_x(x,y) = c_1 + c_2 x + c_3 y + c_4 xy + c_5 x^2 + c_6 y^2 \\ \qquad\qquad + c_7 x^2 y + c_8 x y^2 + c_9 x^3 + c_{10} y^3 \\ T_y(x,y) = c_{11} + c_{12} x + c_{13} y + c_{14} xy + c_{15} x^2 + c_{16} y^2 \\ \qquad\qquad + c_{17} x^2 y + c_{18} x y^2 + c_{19} x^3 + c_{20} y^3 \end{cases} \quad (5)$$

- 上述公式都可以用線性聯立方程式 (system of linear equations) 估計參數。以公式 (4) 為例，首先我們要從兩張影像中找出至少 6 對的對應點 $\{(x_i, y_i), (x_i', y_i'), i = 1, 2, .., 6\}$。我們只有扭曲影像 $g(x', y')$，沒有正常影像 $f(x,y)$；所以正常影像的對應點座標可以來自於已知的校正板 (calibration plate) 或根據已知的點座標關係自行假設。

最小平方誤差估計法 (least-squares estimation)

6 對對應點座標可塑造成 12 個線性方程式

$$\begin{cases} x'_i = T_x(x_i, y_i) = c_1 + c_2 x_i + c_3 y_i + c_4 x_i y_i + c_5 x_i^2 + c_6 y_i^2 \\ y'_i = T_y(x_i, y_i) = c_7 + c_8 x_i + c_9 y_i + c_{10} x_i y_i + c_{11} x_i^2 + c_{12} y_i^2 \end{cases} \tag{6}$$

將這 12 個方程式形成一個矩陣方程式

$$\begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \vdots \\ \vdots \\ x'_6 \\ y'_6 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2 y_2 & x_2^2 & y_2^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_2 & y_2 & x_2 y_2 & x_2^2 & y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_6 & y_6 & x_6 y_6 & x_6^2 & y_6^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_6 & y_6 & x_6 y_6 & x_6^2 & y_6^2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ \vdots \\ \vdots \\ c_{11} \\ c_{12} \end{bmatrix} \tag{7}$$

公式 (7) 矩陣方程式可以簡單表示成

$$\boldsymbol{x}_{12\times1} = \boldsymbol{A}_{12\times12}\, \boldsymbol{c}_{12\times1}, \tag{8}$$

則未知參數向量

$$\boldsymbol{c} = \boldsymbol{A}^{-1} \boldsymbol{x}; \tag{9}$$

如果 6 對對應點共線的話，則 $\boldsymbol{A}$ 是不可逆的。

理論上，用來估計變形公式 (8) 的對應點數愈多，估計愈準確。如果用了 n 對不共線的對應點，則公式 (8) 將變成 $\boldsymbol{x}_{2n\times1} = \boldsymbol{A}_{2n\times12}\, \boldsymbol{c}_{12\times1}$， (10)

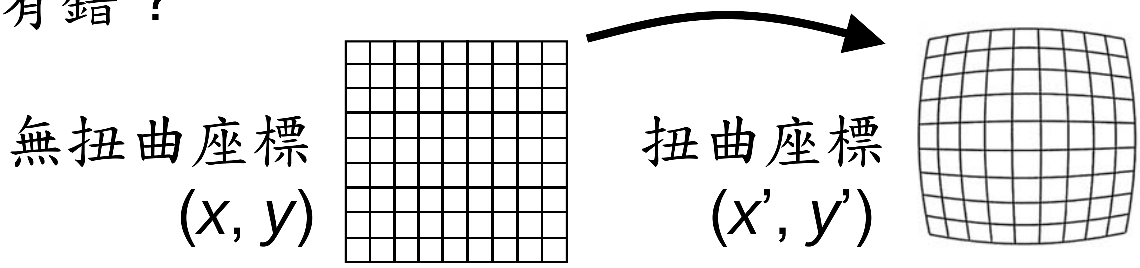
且要用最小平方估計法 (least-squares estimation) 求參數解

$$\boldsymbol{c} = (\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A})^{-1} \boldsymbol{A}^{\mathrm{T}} \boldsymbol{x}, \tag{11}$$

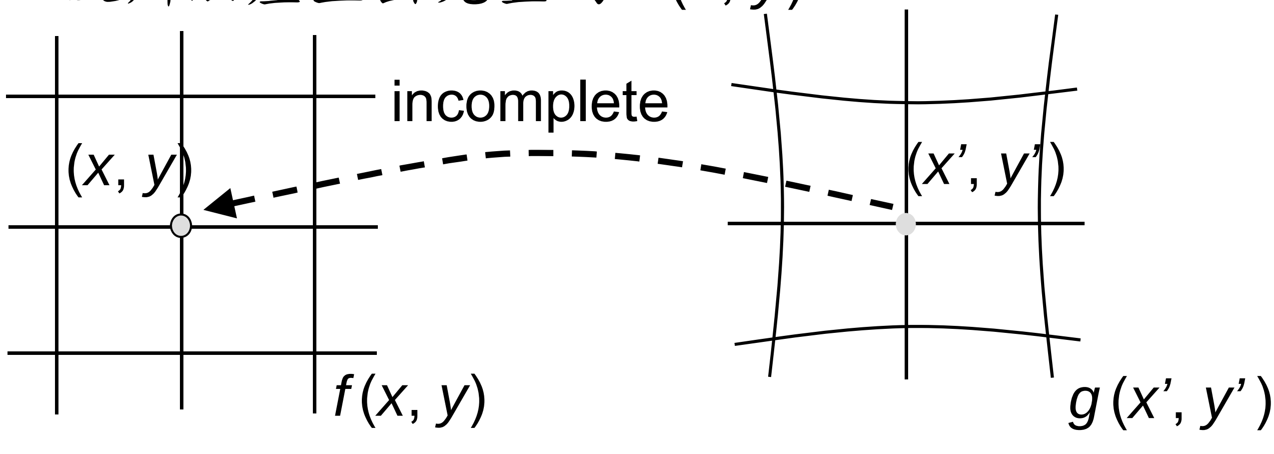
其中 $(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A})^{-1} \boldsymbol{A}^{\mathrm{T}}$ 稱為 $\boldsymbol{A}$ 的假反矩陣 (pseudo-inverse matrix)。

2.6.3 反轉換 (inverse transformation)

- 在上述的二維基本幾何轉換及二維扭曲轉換中，有二個疑問。
- 第一，上述的轉換關係好像方向顛倒了？我們是要從轉換前或扭曲的影像點座標 (x', y') 轉換出轉換後或無扭曲的影像點座標 (x, y)；怎麼上述的轉換公式是從 (x, y) 座標轉換成 (x', y') 座標，有沒有錯？

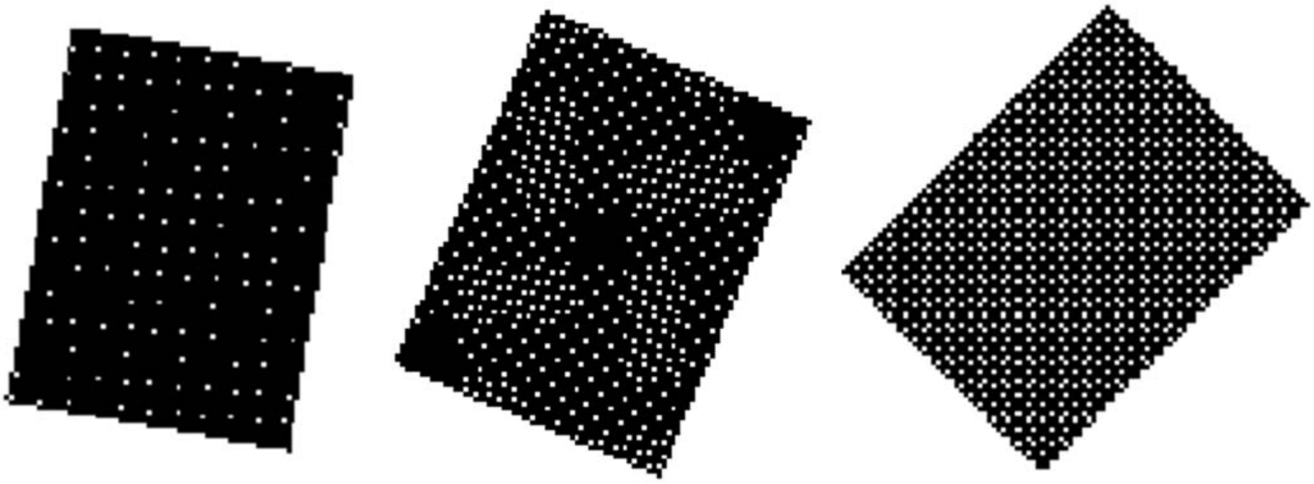


- 其實從扭曲的座標點 (x', y') 計算出無扭曲的座標點 (x, y)；再將扭曲座標點的灰階 $g(x', y')$ 內插出無扭曲影像座標點的灰階 $f(x, y)$；反而才有錯。
- 若依序從每個 (x', y') 計算出相對應的 (x, y)；則有可能某些 (x, y) 沒被對應到，而讓影像出現破洞，沒辦法產生出完整的 $f(x, y)$。



無扭曲影像 (目標影像)　　扭曲影像 (轉換影像)

- 以一張40×30白色影像旋轉為例，計算出新座標後取四捨五入。影像中的黑點表示在目標影像中沒有被對應到的座標。

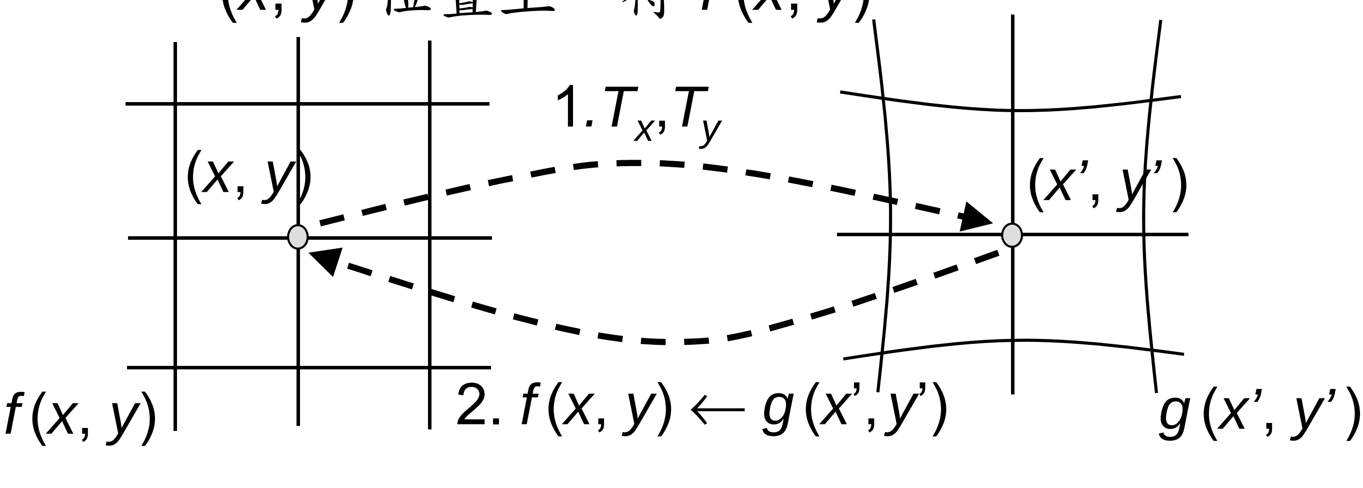


向右旋轉 10°。 向右旋轉 23°。 向右旋轉 45°。

- 若改成依序從每一個 (x, y) 反轉換計算出相對應的 (x', y')；則每一個 (x, y) 都一定會被對應到。

  步驟1. 從目標影像中依序拿出一個座標點 (x, y)；執行座標位置反轉換，得一座標點 (x', y')

  步驟2. 以一個灰階內插法，從 (x', y') 座標點內插出一個灰階 g，再將 g 複製到目標影像的 (x, y) 位置上，得 $f(x, y)$。



- 上述的正反轉換還有另一個疑問，就是如果從整數的 (x, y) 算出來的 (x', y') 不是整數的話怎麼辦 (從整數的 (x', y') 算出來的 (x, y) 也有一樣的問題)？
- 因為這個疑問，所以在上述演算法的步驟 2 中才會有一個“灰階內插”的動作。

註：

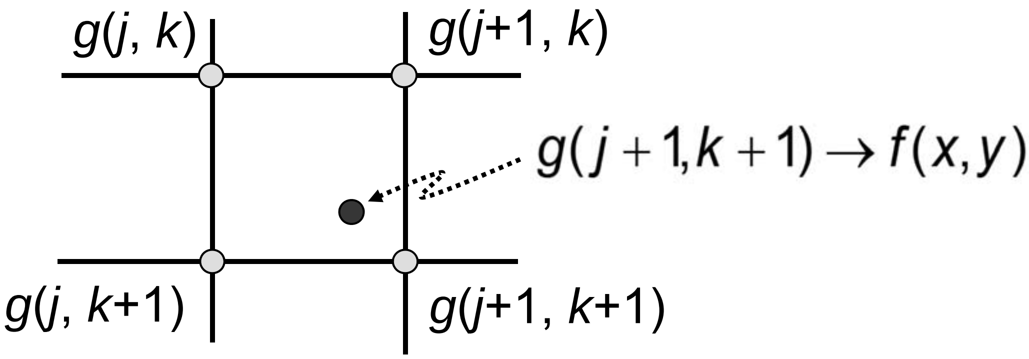
從扭曲影像 (x', y') 轉到正常影像 (x, y) 稱為正轉換
從正常影像 (x, y) 轉到扭曲影像 (x', y') 稱為反轉換

2.6.4 灰階內插

- 三種灰階內插方法

A. 最近鄰點法 (nearest-neighbor approach)

這是最簡單的方法，直接將 (x', y') 四捨五入成 (round (x'), round (y'))，再做 $f(x, y) \leftarrow g$ (round (x'), round (y'))。

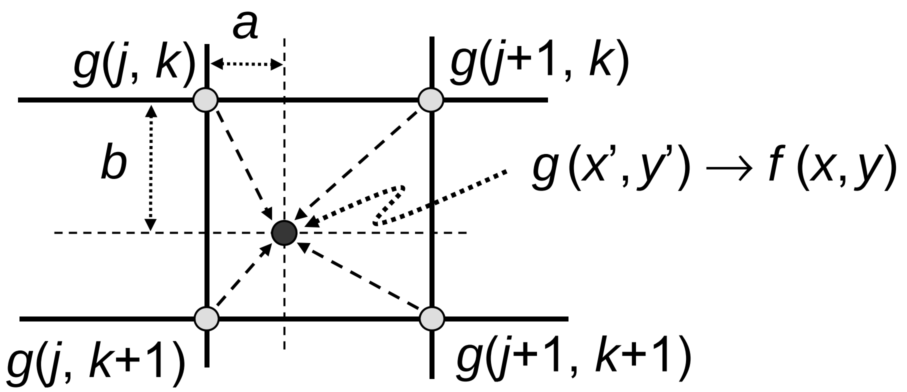


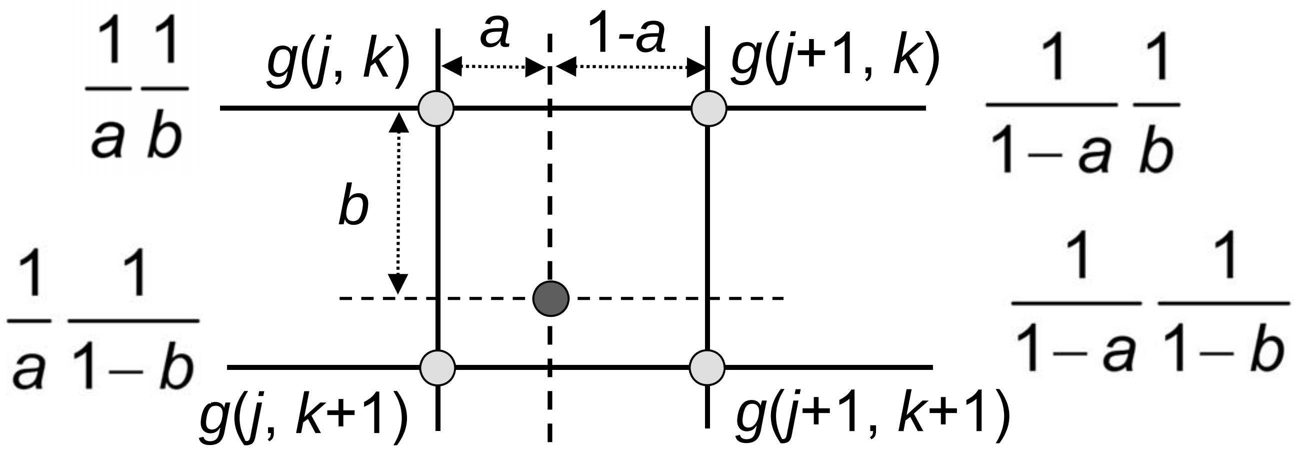
B. 雙線性內插 (bi-linear interpolation)

在扭曲影像中，選取包圍 (x', y') 的四個整數座標點 (j, k), $(j+1, k)$, $(j, k+1)$, 及 $(j+1, k+1)$。再以這四點到 (x', y') 距離的倒數乘以 a b $(1-a)$ $(1-b)$ 當加權數，從扭曲影像的灰階內插出正常影像的灰階，

$$f(x,y) = (1-a)(1-b)\,g(j,k) + a(1-b)\,g(j+1,k) + (1-a)\,b\,g(j,k+1) + a\,b\,g(j+1,k+1),$$

其中 $j = \text{round}(x')$, $k = \text{round}(y')$, $a = x' - j$, 及 $b = y' - k$。





周邊像素各別計算水平或垂直方向的權重。

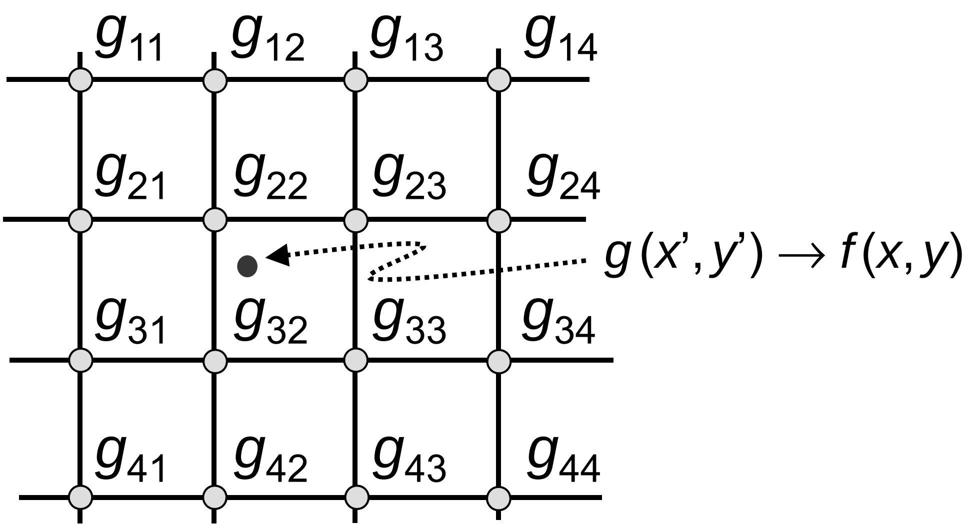
注意：距離倒數當權重，且權重總和要為1；

例如，$g(j, k)$ 與 $g(j, k+1)$ 的水平權重都是

$$\frac{\frac{1}{a}}{\frac{1}{a}+\frac{1}{1-a}} = \frac{\frac{1}{a}}{\frac{(1-a)+a}{a(1-a)}} = \frac{\frac{1}{a}}{\frac{1}{a(1-a)}} = 1-a$$

C. 雙三次曲線內插 (bi-cubic interpolation)

在扭曲影像中，選取包圍 (x',y') 的 16 個整數座標點。再以一維墨西哥帽函數 (Mexican hat function) 定義這 16 點對於 (x', y') 的加權數，從扭曲影像的灰階內插出正常影像的灰階。



為了簡化公式，重新定義座標表示式；
座標 (round(x') -1, round(y') -1) 設為 (1, 1)、(round(x'), round(y')) 設為 (2, 2)，其餘依此類推。所以包圍 (x', y') 的 16 個整數座標點為 (j, k), j, k = 1, .., 4。x 及 y 方向的一維墨西哥帽函數個別定義為

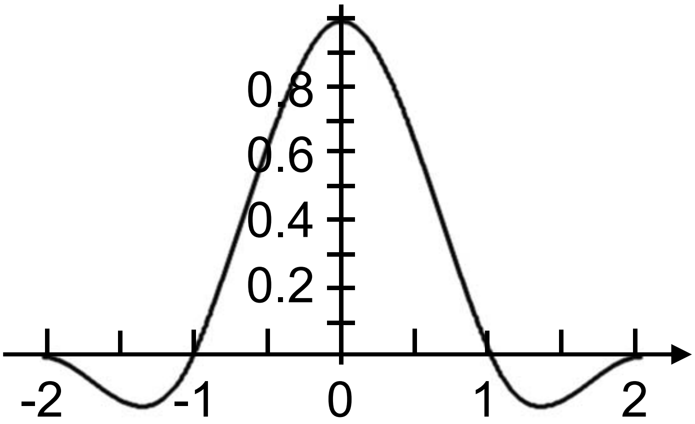
$$w_j = \begin{cases} 1-2|j-x'|^2+|j-x'|^3, & \text{if } 0 \le |j-x'| < 1 \\ 4-8|j-x'|+5|j-x'|^2-|j-x'|^3, & \text{if } 1 \le |j-x'| < 2, \\ 0, & \text{otherwise} \end{cases}$$

$$w_k = \begin{cases} 1-2|k-y'|^2+|k-y'|^3, & \text{if } 0 \le |k-y'| < 1 \\ 4-8|k-y'|+5|k-y'|^2-|k-y'|^3, & \text{if } 1 \le |k-y'| < 2, \\ 0, & \text{otherwise} \end{cases}$$

因此雙三次曲線內插公式則為

$$f(x, y) = \sum_{j=1}^{4} \sum_{k=1}^{4} w_j w_k \, g(j, k).$$

- 一維墨西哥帽函數圖形



- 三種灰階內插的比較，以一個18×18白色方塊向左旋轉 23 o為例



最近鄰點法。 雙線性內插法。 雙三次曲線內插法